**CSCI 6628 OOA/OOD Final Exam Study Guide**

Topics covered 1st half:

1. Vision Statement
2. Product Backlog
3. Use Cases
4. Class Diagrams
5. Interaction Diagrams

Topics covered 2nd half:

1. State Diagrams, Events, Transitions
2. SOA Patterns: Request/Response, Fire and Forget, Message Correlation. Event Processing: Publish/Subscribe. SOA Actor Roles: Consumer & Provider.
3. Introduction to OOD patterns. Assign class responsibilities Interaction and Object design. Learn to apply the GRASP patterns/
4. GoF Patterns: Creational, Structural, Behavioral
5. Domain Model Refinement & Test-Driven Development
6. Design Patterns, Refactoring & Bad Smells
7. Object-Oriented Databases –We talked about only briefly. -No questions on exam about this.
8. You have just been hired, on a temporary trial basis (since yesterday you just graduated from UNH), to run the IT Department for Company ABC. The person that previously had your job was terminated due to not having any software design (UML) artifacts for this project. This virtually guaranteed that the system is unstable, unmaintainable, unreliable and not up to Object-Oriented design best practices. If successful your temporary position will be made permanent, you will be promoted to Director and given a budget which includes the hiring of additional staff.

Using the following screen shots, for the four (4) various menu selections, generate the UML Software Design Artifacts:

* 1. Class Diagram(s)
  2. Interaction/Sequence Diagram(s)
  3. State Diagram(s)

Operational Details

This software is running the … (details and screen shots will be included on Final Exam)

1. Service Oriented Architecture (SOA) Patterns.

Re-do the solution, for problem 1 above, to use SOA Patterns. Pick at least x (the x will be filled-in on the Final Exam) of the following patterns: Request/Response, Request/Reaction, Fire-and-forget, Composite Service Pattern.

1. GRASP Patterns

Re-do the solution, for problem 1 above, to use GRASP patterns. Pick at least x of the following patterns: Creator, Information Expert, Low Coupling, Controller, High Cohesion, Indirection, Polymorphism, Protected Variations, Pure Fabrication.

1. GoF Patterns.

Re-do the solution, for problem 1 above, to use GoF patterns. Pick at least x of the following patterns:

* Singleton
* Simple Factory
* Abstract Factory
* Builder
* Prototype
* Lazy initialization

1. Domain Model Refinement

Modify the solution in problem 1 to utilize Generalization (inheritance) and Association Classes.